Charm++ with UCX: Updates

UCX Dev Workshop 2020

Nitin Bhat, Charmworks Inc.
Jaemin Choi, University of Illinois Urbana Champaign

Outline

1. Charm++ Overview
2. Charm++ and UCX - A brief history
3. Frontera Bugs with Charm/UCX
4. Results
5. Conclusion and Future Work

What is Charm++?

- Charm++ is a generalized approach to writing parallel programs
  - An alternative to the likes of MPI, UPC, GA etc
  - But not to sequential languages as C, C++, and Fortran
- Three key design principles
  - Overdecomposition
  - Migratibility
  - Asynchrony
- Enables features like
  - Automatic overlap of computation and communication
  - Load Balancing
  - Shrink Expand
  - Fault Tolerance
- Represents:
  - The style of writing parallel programs
  - The runtime system
  - And the entire ecosystem that surrounds it

Charm++ and UCX - History

- Why we needed a new communication layer?
  - Verbs was difficult to maintain and not working on new generation of Infiniband machines
  - MPI layer wasn't scaling very well
  - UCX offered portability, high performance and ease of maintenance
- UCX networking layer was added to Charm++ in June 2019
- Nightly build starting July 2019
- Very good initial performance results
  - Pingpong - upto 67% better than MPI, 87% better than Verbs
  - NAMD - 4% better than MPI (22 nodes of Thor)
  - ChaNGa - 37% better than MPI (64 nodes of Frontera)
- Discovery of bugs on Frontera in Nov 2019 - hangs/crashes
- Hangs fixed with the release of UCX v1.9.0-rc1 in Sept 2020, finally!

Frontera Bugs with Charm/UCX

- Applications seeing hangs on Frontera
  - Enzo-P in the non-smp mode hung inconsistently on 4, 8 and 64 node runs
  - NAMD zika virus simulation in the smp mode hung consistently on 16 node runs
  - ChaNGa seeing frequent hangs and crashes on 32/64 node runs
- Tricky bug
  - Couldn’t be reproduced with simpler test cases
  - Couldn’t be reproduced on other machines (Golub/Thor)
- Attempts to try out other layers like MPI
  - Didn’t see any hang with Intel-MPI
  - For custom built MPIs (MPICH, OpenMPI) did see hangs during initialization

Frontera Bugs with Charm/UCX - Debugging attempts

- Had a few discussions with the UCX and NAMD teams.

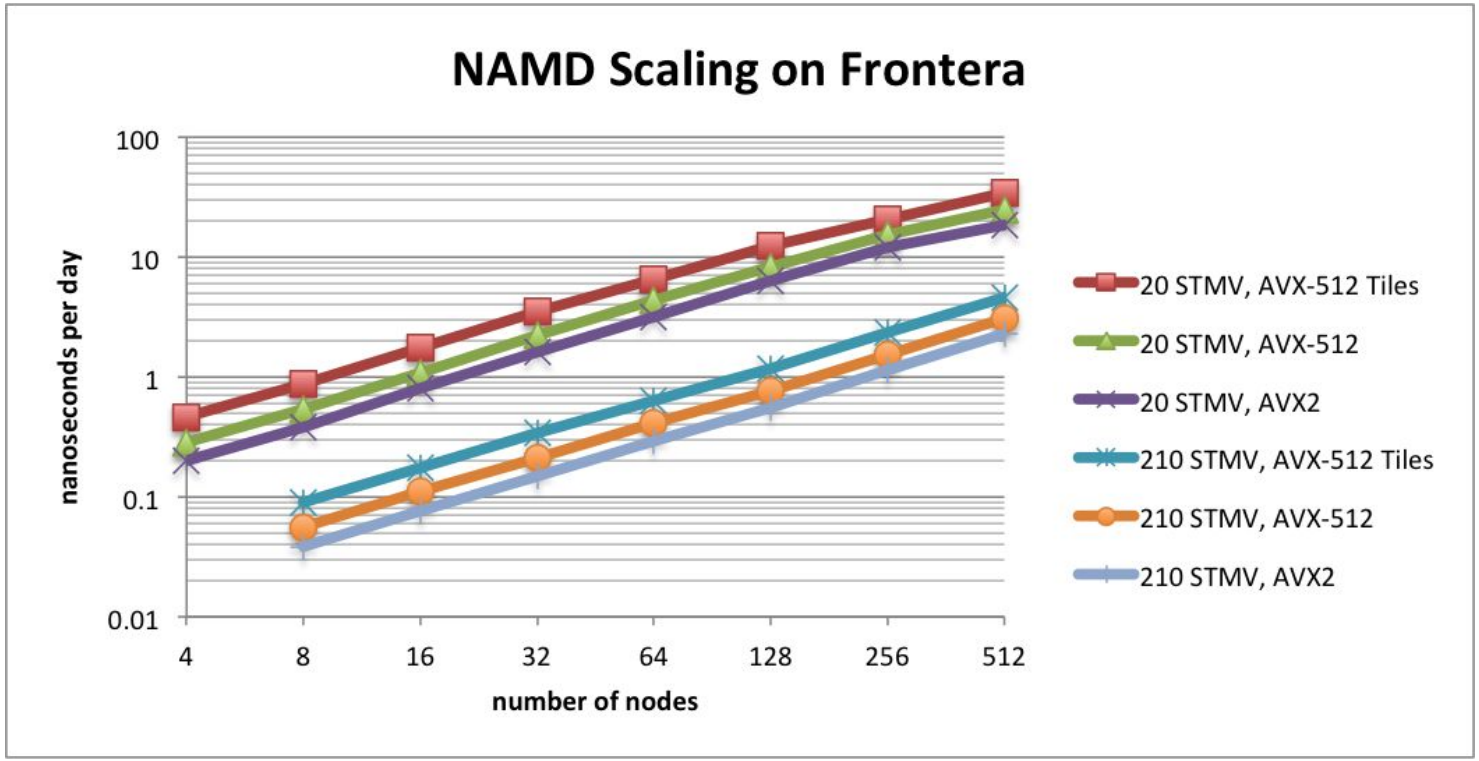
- Getting stack traces from hung processes didn’t reveal anything much, except that Charm’s scheduler loop was executing on all PEs.

- With the arrival of Covid-19, NAMD’s use became even more important
  - Got a special allocation on Frontera for debugging

Frontera Bugs with Charm/UCX - Debugging attempts

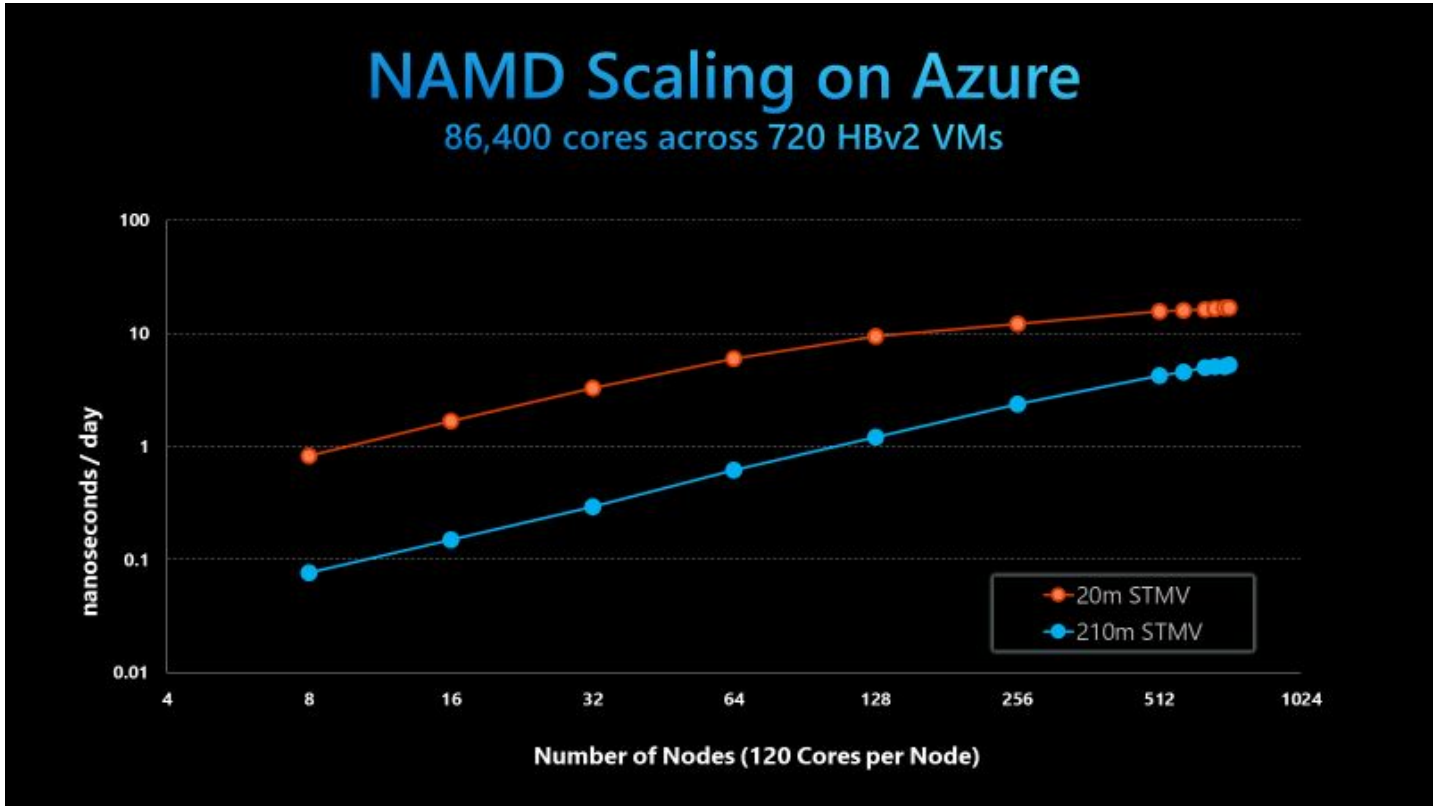
- Lead to development of a message tracking infrastructure
  - Implementation
    - For every message sent, a unique tag per PE is generated and stored
    - For every message received, an ack message is sent back to that PE
    - On receiving the ack, the unique entry is removed
    - When the application hangs, a idle counter triggers a reduction to print out the remaining entries across all PEs.
  - Running with Enzo-P showed that there were unacked messages for a few PEs and showed that the hang was contributed by undelivered messages
- Trying with ucx-master around May/June 2020 didn’t cause the hang to show up anymore. This was a part of ucx 1.9.0 release.

Results from NAMD



https://www.hpcwire.com/2020/08/12/intel-speeds-namd-by-1-8x-saves-xeon-processor-users-millions-of-compute-hours/

Results from NAMD



https://www.hpcwire.com/2020/11/20/azure-scaled-to-record-86400-cores-for-molecular-dynamics/

Frontera Bugs with Charm/UCX - Still Pending

- ChaNGa smp crashes on Frontera with **UCX failed to register user buffer** (https://github.com/UIUC-PPL/charm/issues/2636)
- Similar to Issue opened on ucx repo: https://github.com/openucx/ucx/issues/5291
- Using Active Messaging API seems to be helping avoid the registration issue, runs upto 24 nodes.
- However, 2 node runs still crashes with the same error.

Conclusion and Future Work

- Frontera bugs were very tricky and took a lot of time and effort.
- UCX has proved to be vital for Charm++ to scale well on leading Infiniband machines.
- Thanks to Mikhail for all the help over the year!
- Future Work
  - Performance results from other applications
    - Enzo
    - ChaNGa
  - Using ucx target on other networks
    - Gemini and Slingshot interconnect
    - Pami
  - Features
    - Collectives API
    - Active Messages API (In progress)
    - Inter-GPU communication in Charm++

Questions?

Reach out to me at: nitin@hpccharm.com

Another minor issue - Running without launcher

- On other charm++ layers, single process launches don’t need a launcher. However, on UCX,

- To maintain this uniformity, it’ll be good to have UCX programs also not require a launcher. Currently, it crashes during initialization. (https://github.com/UIUC-PPL/charm/issues/2477)